Comparison of the Worst and Best Sum-Of-Products Expressions for Multiple-Valued Functions

Tsutomu Sasao

Department of Computer Science
and Electronics
Kyushu Institute of Technology
Iizuka 820, Japan

Jon T. Butler

Department of Electrical and
Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121, U.S.A.

Abstract

Because most practical logic design algorithms produce irredundant sum-of-products (ISOP) expressions, the understanding of ISOPs is crucial. We show a class of functions for which Morreale-Minato's ISOP generation algorithm produces worst ISOPs (WSOP), ISOPs with the most product terms. We show this class has the property that the ratio of the number of products in the WSOP to the number in the minimum ISOP (MSOP) is arbitrarily large when the number of variables is unbounded. The ramifications of this are significant; care must be exercised in designing algorithms that produce ISOPs.

We also show that 2^{n-1} is a (firm) upper bound on the number of product terms in any ISOP for switching functions on n variables, answering a question that has been open for 30 years. We show experimental data and extend our results to functions of multiple-valued variables.

Index terms: Logic minimization, irredundant sum-of-products, multiple-valued logic.

1 Introduction

The majority of logic minimization algorithms used in practical design produce irredundant sum-of-products expressions (ISOPs) rather than minimum sum-of-products expressions (MSOPs). For example, the PRESTO [2, 21] logic minimizer produces an ISOP as follows:

1) Expand each product into a prime implicant
2) Delete redundant prime implicants.

An ISOP is the OR of prime implicants such that deleting any prime implicant changes the function. For example, two expressions $x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee \bar{x}_1 x_3$ and $x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee \bar{x}_1 x_2 \vee \bar{x}_1 x_3$ are both ISOPs for the same function (See Fig 1(a) and (b)). However, only the former is an MSOP. Depending on one's viewpoint, the second ISOP has only one more product than the first, or 33% more products.

These two viewpoints inspire the first question: "Can a logic minimization algorithm yield unreasonably large SOPs?" The answer is yes. We show that there exists an algorithm [13, 14] that produces worst ISOPs (WSOPs), ISOPs with the largest number of product terms, for some class of functions.

The question above and its surprising answer inspire the second question: "To what extent can the number of products in a WSOP exceed the number of products in an MSOP?" The answer to this is also surprising. We show there exist functions in which the ratio of the WSOP product count to MSOP product count is arbitarily large, when the number of variables is unbounded.

While our results are motivated by the existence of binary logic functions, we show that a similar phenomenon exists for functions of a higher radix. Our results suggest that the disparity between the number of products in multiple-valued WSOPs and MSOPs increases with radix.

2 Definitions and Basic Properties

Definition 2.1 *x and $\bar{x}$ are literals of a variable x. A logical product that contains at most one literal for each variable is called a product term or a product. Products combined with OR operators form a sum-of-products expression (SOP).*

Definition 2.2 *A prime implicant (PI) of a function f is a product which implies f, such that the deletion of any literal from the product results in a new product that does not imply f.*

Definition 2.3 *An irredundant sum-of-products expression (ISOP) is an SOP, where each product is a PI, and no product can be deleted without changing the function represented by the expression.*

Definition 2.4 *Among the ISOPs for f, one with the maximum number of products is a worst ISOP (WSOP), and one with the minimum number of products is a minimal SOP (MSOP).*

Definition 2.5 *The number of products in a WSOP for f is denoted by $\tau(WSOP : f)$. The number of products in an MSOP for f is denoted by $\tau(MSOP : f)$.*

0-8186-7910-7/97 $10.00 © 1997 IEEE

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **MAY 1997**	2. REPORT TYPE	3. DATES COVERED

4. TITLE AND SUBTITLE **Comparison of the Worst and Best Sum-Of-Products Expressions for Mult iple-Valued Functions**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Electrical and Computer Engineering,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Because most practical logic design algorithms produce irredundant sum-of-products (ISOP) expressions, the understanding of ISOPs is crucial. We show a class of functions for which Morreale-Minato ?s ISOP generation algorithm produces worst ISOPs (WSOP), ISOPs with the most product terms. We show this class has the property that the ratio of the number of products in the WSOP to the number in the minimum ISOP (MSOP) is arbitrarily large when the number of variables is unbounded. The ramifications of this are significant; care must be exercised in designing algorithms that produce ISOPs. We also show that 2?-? is a (firm) upper bound on the number of product terms in any ISOP for switching functions on n variables, answering a question that has been open for 30 years. We show experimental data and extend our results to functions of multiple-valued variables. Index terms: Logic minimization, irredundant sum-ofproducts multiple-valued logic.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

The following is well known [7, 15]

Theorem 2.1 *For any switching function of n variables, $\tau(MSOP : f) \leq 2^{n-1}$.*

Our first result shows that Theorem 2.1 is also true when we replace MSOP by WSOP. It answers an open question posed by Meo [10] in 1968.

Theorem 2.2 *For any switching function of n variables, $\tau(WSOP : f) \leq 2^{n-1}$.*

(Proof) Available form the authors.

There exists a WSOP with 2^{n-1} products; it is the SOP of a parity function of n variables. Thus, the upper bound in Theorem 2.2 is tight.

Lemma 2.1 *Let $g(X)$ and $h(Y)$ be functions, where X and Y have no common variables. Let $G(X)$ and $H(Y)$ be ISOPs for $g(X)$ and $h(Y)$, respectively. Then, the SOP $F(X,Y)$ derived from $G(X)H(Y)$ by using distributive laws is an ISOP for f.*

(Proof) Clearly, $F(X,Y)$ represents f. If any product in $F(X,Y)$ is not a PI, then either $G(X)$ or $H(Y)$ or both contain a non-prime product, which contradicts the assumption that $G(X)$ and $H(Y)$ are ISOPs. If any product in $F(X,Y)$ is redundant, then either $G(X)$ or $H(Y)$ or both contain a redundant product which contradicts the assumption that $G(X)$ and $H(Y)$ are irredundant. ∎

Our next result shows that if a function $f(X,Y)$ can be expressed as the AND of two functions, $g(X)$ and $h(Y)$ on disjoint sets of variables, then the number of product terms in a WSOP (MSOP) of $f(X,Y)$ is the product of the number of products in a WSOP (MSOP) of $g(X)$ and a WSOP (MSOP) of $h(Y)$.

Theorem 2.3 *Let $g(X)$ and $h(Y)$ be functions, where X and Y have no common variables. Let $f(X,Y) = g(X)h(Y)$. Then,*

1. *$\tau(WSOP : f) = \tau(WSOP : g)\tau(WSOP : h)$,*
2. *$\tau(MSOP : f) = \tau(MSOP : g)\tau(MSOP : h)$.*

(Proof) Available form the authors.

This result will be useful later when we demonstrate functions with a large discrepancy between the number of products in the WSOP and in the MSOP.

3 Comparing the Number of Product Terms in a WSOP to the Number in an MSOP for Specific Functions

Definition 3.6 *Let $ST(n,k)$ be a symmetric function of n-variables $x_1, x_2, \ldots, x_n$ such that*

$$ST(n,k) = \begin{cases} 1 & k \leq \sum_{i=1}^{n} x_i \leq n-k \\ 0 & \textit{otherwise}, \end{cases}$$

where $\sum$ is ordinary addition in which the value of x_i is viewed as an integer. That is, $\sum_{i=1}^{n} x_i$ is the number of variables that are 1.

Example 3.1 *$ST(n,0) = 1$. $ST(n, \frac{n}{2})$, for even n, is the OR of all minterms with exactly half of the variables complemented.* *(End of Example)*

Lemma 3.2 *$ST(n,k)$ can be represented as*

$$ST(n,k) = S_{\{k,k+1,\ldots,n\}} S_{\{0,1,\ldots,n-k\}},$$

where

$$S_A = \begin{cases} 1 & \textit{if} \left(\sum_{i=1}^{n} x_i \right) \in A \\ 0 & \textit{otherwise}, \end{cases}$$

where $A \subseteq \{0, 1, 2, \ldots, n\}$.

Example 3.2

$$\begin{aligned} ST(n,1) &= S_{\{1,2,\ldots,n\}} S_{\{0,1,\ldots,n-1\}} \\ &= (x_1 \vee x_2 \vee \cdots \vee x_n)(\bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_n) \end{aligned}$$

and

$$\begin{aligned} ST(n,2) &= S_{\{2,3,\ldots,n\}} S_{\{0,1,\ldots,n-2\}} \\ &= (x_1x_2 \vee x_1x_3 \vee \cdots \vee x_{n-1}x_n) \\ &\quad (\bar{x}_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3 \vee \cdots \vee \bar{x}_{n-1}\bar{x}_n). \end{aligned}$$

(End of Example)

We are interested in the *total* number of PIs in $ST(n,k)$, the number of PIs in an MSOP for $ST(n,k)$, and the number of PIs in a WSOP for $ST(n,k)$. We can derive these as follows.

Theorem 3.4

1) *$ST(n,k)$ has $\binom{n}{k,\, n-2k,\, k} = \frac{n!}{k!(n-2k)!k!}$ PIs.*
2) *$\tau(MSOP : ST(n,k)) = \binom{n}{k} = \frac{n!}{k!(n-k)!}$.*
3) *$\tau(WSOP : ST(n,k)) = 2\binom{n}{k} - \binom{2k}{k}$.*

(Proof) Available form the authors.

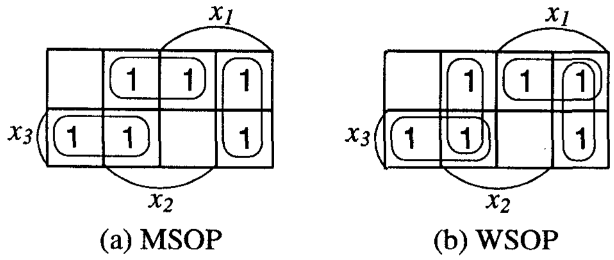
A special case of Theorem 3.4 occurs when $n = 3$ and $k = 1$.

Corollary 3.1 *$ST(3,1)(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$ has the following properties:*

1) *$ST(3,1)$ has 6 PIs.*
2) *$\tau(MSOP : ST(3,1)) = 3$.*
3) *$\tau(WSOP : ST(3,1)) = 4$.*

Fig. 1 (a) and (b) suggest how the MSOP and WSOP are formed. With the MSOP, product terms cover as many minterms as possible. With the WSOP, product terms are chosen so that as much overlap occurs as possible. Notice that any product term added to Fig. 1(a) or (b) is redundant. We now consider the questions posed in the introduction.

Definition 3.7 *A set of true minterms S for f is called independent if no implicant of f contains a pair of minterms in S.*



(a) MSOP (b) WSOP

Figure 1. Karnaugh maps for $ST(3,1)$.

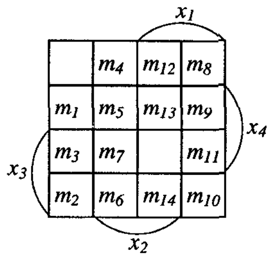


Figure 2. Map for $ST(4,1)$.

Lemma 3.3 *Let S be an independent set of minterms for f. Then $\tau(MSOP : f) \geq |S|$.*

Example 3.3 *Consider the Karnaugh map of $ST(4,1)$ in Fig. 2. Independent sets for $ST(4,1)$ include $S_1 = \{m_7, m_{11}, m_{13}, m_{14}\}$ and $S_2 = \{m_1, m_2, m_4, m_8\}$. From Lemma 3.3, $\tau(MSOP : ST(4,1)) \geq 4$. (End of Example)*

Definition 3.8 *The redundancy ratio of a function f is*

$$\rho(f) = \frac{\tau(WSOP : f)}{\tau(MSOP : f)}.$$

The normalized redundancy ratio of an n-variable function f is

$$\sigma(f) = \sqrt[n]{\rho(f)}.$$

The redundancy ratio is a measure of the discrepancy between the number of product terms in WSOPs and MSOPs. A small ratio suggests that any logic minimization algorithm will do well, while a large ratio suggests that care should be exercised. The normalized redundancy ratio is normalized with respect to the number of variables. It is a convenience; it allows one to compare the redundancy ratio of two functions with a different number of variables.

Example 3.4

$$\begin{array}{lll}
\rho(ST(3,1)) = \frac{4}{3}, & \sigma(ST(3,1)) = \sqrt[3]{\frac{4}{3}} & \simeq 1.1006 \\
\rho(ST(4,1)) = \frac{6}{4}, & \sigma(ST(4,1)) = \sqrt[4]{\frac{6}{4}} & \simeq 1.1066 \\
\rho(ST(5,1)) = \frac{8}{5}, & \sigma(ST(5,1)) = \sqrt[5]{\frac{8}{5}} & \simeq 1.0986 \\
\rho(ST(6,1)) = \frac{10}{6}, & \sigma(ST(6,1)) = \sqrt[6]{\frac{10}{6}} & \simeq 1.0889 \\
\rho(ST(7,1)) = \frac{12}{7}, & \sigma(ST(7,1)) = \sqrt[7]{\frac{12}{7}} & \simeq 1.0800 \\
\rho(ST(8,1)) = \frac{14}{8}, & \sigma(ST(8,1)) = \sqrt[8]{\frac{14}{8}} & \simeq 1.0725.
\end{array}$$

(End of Example)

For these $ST(n,1)$ functions, σ is the largest when $n = 4$. That is, as n increases above 3, σ first increases peaking at 4, and then it continually decreases.

To understand how poorly an ISOP generator can do, we investigate functions with large ρ. For such functions, the choice of algorithm is important; a poorly designed algorithm can produce ISOPs with many products. When ρ is small, there is less concern; any ISOP generator algorithm will do well. Our first result below shows that the $ST(n,k)$ functions have reasonably small ρ.

Theorem 3.5

$$1 \leq \rho(ST(n,k)) < 2.$$

(Proof) The first inequality follows from the fact that a WSOP has at least as many product terms as an MSOP. The second inequality is proved as follows.

$$\begin{aligned}
\rho(ST(n,k)) &= \frac{\tau(ST(n,k) : WSOP)}{\tau(ST(n,k) : MSOP)} \\
&= \frac{2\binom{n}{k} - \binom{2k}{k}}{\binom{n}{k}} = 2 - \frac{\binom{2k}{k}}{\binom{n}{k}}.
\end{aligned}$$

Since $\frac{\binom{2k}{k}}{\binom{n}{k}} > 0$, we have the theorem. ∎

Note that the largest ρ is achieved when n is much larger than k, and this value can never be more than 2. However, one can use $ST(n,k)$ functions to construct functions with large ρ, as follows.

Definition 3.9 *Let $ST(n,k)^r$ be the $n \cdot r$-variable function*

$$\begin{aligned}
&ST(n,k)^r(x_1, x_2, \ldots, x_{nr}) \\
&= \bigwedge_{i=1}^{r} ST(n,k)(x_{n(i-1)+1}, x_{n(i-1)+2}, \ldots, x_{ni}).
\end{aligned}$$

Lemma 3.4 *Let $g_i(X)$ be a function with redundancy ratio $\rho(g_i)$. Let $f(X_1, X_2, \ldots, X_r) = g_1(X_1)g_2(X_2)\cdots g_r(X_r)$, where $X_1, X_2, \ldots,$ and X_r are pairwise disjoint. Then, $\rho(f) = \prod_{i=1}^{r} \rho(g_i)$.*

Theorem 3.6 *$ST(n,k)^r$ has the following properties:*

1) $ST(n,k)^r$ has $\binom{n}{k,\,n-2k,\,k}^r = (\frac{n!}{k!(n-2k)!k!})^r$ PIs.
2) $\tau(MSOP : ST(n,k)^r) = \binom{n}{k}^r$.
3) $\tau(WSOP : ST(n,k)^r) = [2\binom{n}{k} - \binom{2k}{k}]^r$.

Example 3.5 *For $n = 3$ and $k = 1$, $ST(3,1)^r$ has 6^r PIs, $\tau(MSOP : ST(3,1)^r) = 3^r$, and $\tau(WSOP : ST(3,1)^r) = 4^r$. (End of Example)*

The ISOP generator developed by Minato [13] is quite fast. Unfortunately, it produces WSOPs instead of MSOPs for $ST(3,1)^k$. This heuristic logic minimizer produces ISOPs that have many more products than MSOPs. This answers the first question posed in the introduction.

We have

Theorem 3.7

$$\rho(ST(n,k)^r) = \left[2 - \frac{\binom{2k}{k}}{\binom{n}{k}}\right]^r.$$

Example 3.6 *For $n = 4$ and $k = 1$ we have*

$$\rho(ST(4,1)^r) = (1.5)^r.$$

(End of Example)

From this, it can be seen that ρ can be arbitrarily large. This answers the second question posed in the introduction.

4 Extension to Multiple-Valued Functions

4.1 Multiple-Valued Input Two-valued Output Functions

Definition 4.10 *A multi-valued input two-valued output function is f: $P_1 \times P_2 \times \cdots \times P_n \rightarrow \{0,1\}$, where $P_i = \{0,1,\ldots,p_{i-1}\}$, $p_i \geq 2$.*

Definition 4.11 *X^S is a literal of p-valued variable X, where $S \subseteq \{0,1,\ldots,p-1\}$. $X^S = 1$ if $X = a \in S$ and $X^S = 0$, otherwise. A logical product of literals that contains at most one literal for each variable is a product term or a product. Products combined with OR operators form a sum-of-products expression (SOP). Prime implicants (PI), irredundant sum-of-products expression (ISOP), worst ISOP (WSOP), and minimum SOP (MSOP) are defined in a manner similar to the binary case.*

Theorem 4.8 *([17]) Let f be the function defined in Definition 4.11. Then,*

$$\tau(MSOP : f) \leq B = \frac{\prod_{i=1}^{n} p_i}{\max_{i=1}^{n} p_i}.$$

When $p_i = 2 (i = 1, 2, \ldots, n)$, i.e., for switching functions, $B = 2^{n-1}$. In this case, $\tau(WSOP : f) \leq B$ (Theorem 2.1), and $\tau(WSOP : f) \leq B$ (Theorem 2.2). However, for some functions on multiple-valued variables, an ISOP (e.g. a WSOP) can have more than B products. i.e., $\tau(WSOP : f) > B$.

Example 4.7 *Consider the function f: $P_1 \times P_2 \times P_3 \times P_4 \times P_5 \rightarrow \{0,1\}$, where $P_1 = P_2 = P_3 = P_4 = \{0,1\}$ and $P_5 = \{0,1,2,3\}$. The map is shown in Fig. 3. Since all the implicants are prime and irredundant, the SOP shown in the map is an ISOP. Note that this ISOP requires 17 products. On the other hand, Theorem 4.8 gives $B = 16$. Thus, $\tau(WSOP : f) > B = 16$.* *(End of Example)*

Example 4.8 *Consider the function MVO4: $P_1 \times P_2 \times P_3 \times P_4 \times P_5 \rightarrow \{0,1\}$, where $P_1 = P_2 = P_3 = P_4 = \{0,1\}$ and $P_5 = \{0,1,\ldots,15\}$. Fig. 4 shows the positional cube notation [20] of the expression. The cubes are prime and irredundant, so this figure represents a WSOP with 16 products. Note that the MSOP has only 8 products. (End of Example)*

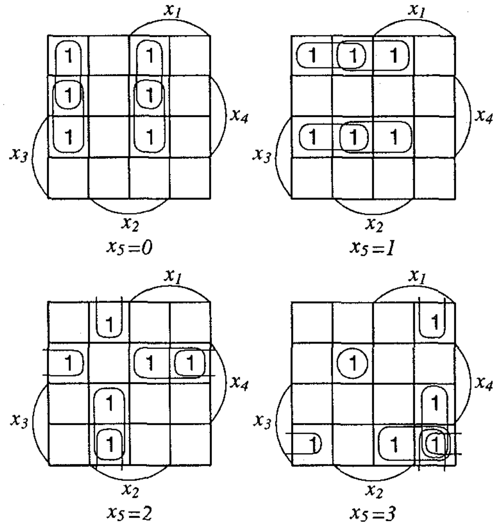


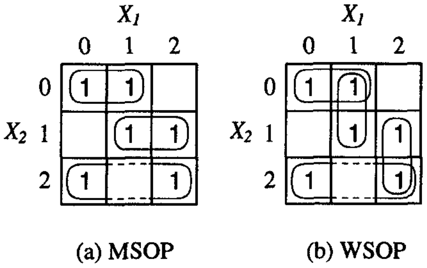
Figure 3. Multiple-valued function.

WSOP				
10	10	10	10	0111111111111111
10	10	10	01	1011111111111111
10	10	01	10	1101111111111111
10	10	01	01	1110111111111111
10	01	10	10	1111011111111111
10	01	10	01	1111101111111111
10	01	01	10	1111110111111111
10	01	01	01	1111111011111111
01	10	10	10	1111111101111111
01	10	10	01	1111111110111111
01	10	01	10	1111111111011111
01	10	01	01	1111111111101111
01	01	10	10	1111111111110111
01	01	10	01	1111111111111011
01	01	01	10	1111111111111101
01	01	01	01	1111111111111110
MSOP				
10	11	11	11	0000000011111111
01	11	11	11	1111111100000000
11	01	11	11	1111000011110000
11	10	11	11	0000111100001111
11	11	10	11	0011001100110011
11	11	01	11	1100110011001100
11	11	11	01	1010101010101010
11	11	11	10	0101010101010101

Figure 4. Arrays for the WSOP and MSOP for MVO4.

By generalizing the above example, we have the following:

Theorem 4.9 *There exists a function MVOn: $P_1 \times P_2 \times \cdots \times P_n \times P_{n+1} \rightarrow \{0,1\}$, where $P_1 = P_2 = \cdots = P_n = \{0,1\}$ and $P_{n+1} = \{0,1,\ldots,2^{n-1}\}$, such that a WSOP requires 2^n products, while an MSOP requires $2n$ products.*



(a) MSOP (b) WSOP

Figure 5. $MV2(X_1, X_2)$.

Definition 4.12 *The function* $MV2$: $\{0,1,2\}^2 \to \{0,1\}$ *is defined as follows:*

$$MV2(X_1, X_2) = X_1^{\{0,1\}}X_2^{\{0\}} \vee X_1^{\{1,2\}}X_2^{\{1\}} \vee X_1^{\{0,2\}}X_2^{\{2\}}.$$

From Fig 5, we have the following:

Lemma 4.5 *$MV2$ has the following properties:*

1) $MV2$ has 6 PIs.
2) $\tau(MSOP : MV2) = 3$.
3) $\tau(WSOP : MV2) = 4$.

Definition 4.13 *A $2k$-variable function $MV2^k$ is defined as follows:*

$$MV2^k(X_1, X_2, \ldots, X_{2k}) = \bigwedge_{i=1}^{k} MV2(X_{2i-1}, X_{2i}).$$

Theorem 4.10 *$MV2^k$ has the following properties:*

1) $MV2^k$ has 6^k PIs.
2) $\tau(MSOP : MV2^k) = 3^k$.
3) $\tau(WSOP : MV2^k) = 4^k$.

Thus, $\rho(MV2) = 4/3$, $\sigma(MV2) = 2/\sqrt{3} = 1.154$.

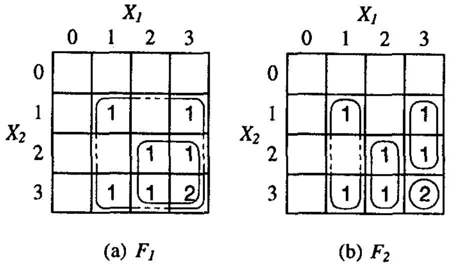
4.2 Multiple-Valued Logic Functions

Definition 4.14 *A multiple-valued logic function is f: $P^n \to P$, where $P = \{0, 1, \ldots, p-1\}$ and $p \geq 3$.*

In the case of multiple-valued logic functions, two different "sum" operators exist: "MAX" and "truncated sum." When the expression uses Max operators, we need only consider the prime implicants. (We assume that the literal takes only two values. If the literal can be any function of one-variable, then minimum SOP may contain non-prime implicant [12]). However, when the expression uses truncated sum, minimum SOPs may contain non-prime implicants, as well as prime implicants [5].

Example 4.9 *Consider the following expressions:*

$$\begin{aligned} F1 &= X_1^{\{1,3\}}X_2^{\{1,3\}} + X_1^{\{2,3\}}X_2^{\{2,3\}} & (1) \\ F2 &= X_1^{\{1\}}X_2^{\{1,3\}} + X_1^{\{2\}}X_2^{\{2,3\}} & \\ &\quad +X_1^{\{3\}}X_2^{\{1,2\}} + 2X_1^{\{3\}}X_2^{\{3\}}. & (2) \end{aligned}$$



(a) F_1 (b) F_2

Figure 6. Expression using truncated sum.

Fig. 6(a) and (b) are maps for (1) and (2), respectively. Note that both F_1 and F_2 are irredundant, and represent the same function. Also, note that F_2 consists of non-prime implicants. In this case, $\tau(F_1) = 2$ and $\tau(F_2) = 4$. Thus, $\rho(f) = 4/2 = 2$ and $\sigma(f) = \sqrt{2}$. (End of Example)

For expressions using truncated sum operators, $\rho(f)$ can be larger than in the binary case. For expressions using MAX operators, the MSOP can be obtained by minimizing expressions for multiple-valued input functions with don't cares [18, 20].

5 Experimental Results and Observations

5.1 Two-valued Case

We generated $ST(n,k)^r$ for different n and k, and obtained their ISOPs. To generate ISOPs, we used Minato's method [13] which is based on Morreale's algorithm [14]. Minato's methods produced WSOPs for all the functions in Table 1.

The 9SYM [6, 19, 22] function shown in page 165 of [1] is identical to $ST(9,3)$. It has 1680 PIs, $\tau(WSOP : 9SYM) = 148$, and $\tau(MSOP : 9SYM) = 84$. POP [3], a PRESTO [2, 21] type logic minimization algorithm, produced a solution with 148 products. Thus, POP produced a WSOP.

5.2 Multiple-Valued Case

For multiple-valued functions, we generated MVO4^2. This function has 6400 PIs. To obtain an ISOP, we used the following method [4, 8, 20]: (note that Minato's method does not apply to the multiple-valued case).

1) Generate the set S of PIs of f.

2) For each cube c in S, do the following: if c is contained by $S - c$, then $S \leftarrow S - c$.

It produced an ISOP with 256 products, which is the WSOP. The MSOP has only 64 products. So, $\rho(MVO4^2) = 4$.

6 Conclusions and Comments

The analysis of ISOPs is important because ISOPs are so often used in logic synthesis. Their importance, however, was recognized 30 years ago when Meo [10] conjectured that

Table 1. Number of products and redundancy ratio for various functions.

	n	MSOP	WSOP	ISOP Minato	PI	ρ	σ
$ST(3,1)$	3	3	4	4	6	1.33333	1.1006
$ST(3,1)^2$	6	9	16	16	36	1.77778	
$ST(3,1)^3$	9	27	64	64	216	2.37037	
$ST(3,1)^4$	12	81	256	256	1296	3.16049	
$ST(3,1)^5$	15	243	1024	1024	7776	4.21399	
$ST(3,1)^6$	18	729	4096	4096	46656	5.61866	
$ST(4,1)$	4	4	6	6	12	1.50000	1.1067
$ST(4,1)^2$	8	16	36	36	144	2.25000	
$ST(4,1)^3$	12	64	216	216	1728	3.37500	
$ST(4,1)^4$	16	256	1296	1296	20736	5.06250	
$ST(4,1)^5$	20	1024	7776	7776	248832	7.59375	
$ST(5,1)$	5	5	8	8	20	1.60000	1.0986
$ST(5,2)$	5	10	14	14	30	1.40000	1.0696
$ST(5,1)^2$	10	25	64	64	400	2.56000	
$ST(5,1)^3$	15	125	512	512	8000	4.09600	
$ST(5,1)^4$	20	625	4096	4096	160000	6.55360	
$ST(6,1)$	6	6	10	10	30	1.66667	1.0889
$ST(6,2)$	6	15	24	24	90	1.60000	1.0815
$ST(7,1)$	7	7	12	12	42	1.71429	1.0800
$ST(7,2)$	7	21	36	36	210	1.71429	1.0800
$ST(7,3)$	7	35	50	50	420	1.42857	1.0523
$ST(8,1)$	8	8	14	14	56	1.75000	1.0725
$ST(8,2)$	8	28	50	50	420	1.78571	1.0752
$ST(8,3)$	8	56	92	92	560	1.64286	1.0640
$ST(9,1)$	9	9	16	16	72	1.77778	1.0660
$ST(9,2)$	9	36	66	66	756	1.83333	1.0697
$ST(9,3)$	9	84	148	148	1680	1.76191	1.0650
$ST(9,4)$	9	126	182	182	630	1.44444	1.0417
$ST(10,1)$	10	10	18	18	90	1.80000	1.0605

n: number of input variables.
MSOP: number of products in MSOP.
WSOP: number of products in WSOP.
ISOP: number of products in ISOP (Minato's method).
PI: number of prime implicants.
ρ: redundancy ratio: $\tau(WSOP : f)/\tau(MSOP : f)$
σ: normalized redundancy ratio: $\sqrt[n]{\rho}$

2^{n-1} was the largest number of products in an ISOP of n-variable functions. In this paper, we settle this open question, showing indeed that 2^{n-1} is firm upper bound.

We also show a class of functions in which an ISOP generation algorithm [13] produces a WSOP, an ISOP with the largest number of products. Such functions are useful for comparing the performance of two-level logic minimizers. We also show, for this class, that as the number of variables increases, ρ, the ratio of the number of products in the WSOP to the number of products in the MSOP increases arbitrarily. These two results clearly show that it is important to develop good minimization algorithms [1, 4, 8, 16]. Specifically, it shows that 1) reasonable algorithms can produce poor results and 2) these poor results can be far from minimum.

A key part of our results is a theorem that allows us to magnify small values of ρ. That is, we compose functions on many variables with large ρ from functions on few variables with small ρ.

We present experimental results and we extend our analysis to multiple-valued logic. For example, we show a multiple-valued function whose WSOP has four times the products than the MSOP.

Acknowledgments

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Culture and Sports of Japan. Discussions with Dr. Shinichi Minato were quite useful.

References

[1] R. K. Brayton, G. D. Hachtel, C. T. McMullen and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Boston, MA., Kluwer Academic Publishers, 1984.

[2] D. W. Brown, "A state-machine Synthesizer –SMS," *Proc. of 18th Design Automation Conference*, June 1981.

[3] G. De Micheli, M. Hofmann, R. Newton, A. Sangivanni-Vincentelli, "A design system for the PLA-based digital circuits," in *Advances in Computer-Aided Engineering Design*, Volume 1, pp. 285-364, Jay Press, 1985.

[4] D. L. Dietmeyer, *Logic Design of Digital Systems* (Second Edition), Allyn and Bacon Inc., Boston, 1978.

[5] G. Dueck and D. M. Miller, "A 4-valued PLA using the MOD SUM," *Proc. of the 16th International Symp. on Multiple-valued Logic*, pp. 232-240, May 1986.

[6] B. Dunham and R. Fridshal, "The problem of simplifying logical expressions," *J. Symbolic Logic 24*, pp. 17-19, 1959.

[7] M. A. Harrison, *Introduction to Switching and Automata Theory*, McGraw-Hill, 1965.

[8] S. J. Hong, R. G. Cain and D. L. Ostapko, "MINI: A heuristic approach for logic minimization," *IBM J. Res. & Develop.*, pp. 443-458, Sept. 1974.

[9] D. E. Knuth, *Fundamental Algorithms*, (The Art of Computer Programming Volume 1), Second Edition, Addison-Wesley, 1973.

[10] A. R. Meo, "On the synthesis of many-variable switching functions," in *Networks and Switching Theory* (G. Biorci, ed.), Chapter VI, pp. 470-482, Academic Press, New york, 1968.

[11] F. Mileto and G. Putzolu, "Average values of quantities appearing in Boolean function minimization," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 81-92, Apr. 1964.

[12] D. M. Miller and J. C. Muzio, "On the minimization of many-valued functions," *Proc. of the 9th International Symposium on Multiple-Valued Logic*, pp. 294-299, May 1979.

[13] S. Minato, "Fast generation of prime-irredundant covers from binary decision diagrams," *IEICE Trans. Fundamentals*, Vol. E76-A, No. 6, pp. 976-973, June 1993.

[14] E. Morreale, "Recursive operators for prime implicant and irredundant normal form determination," *IEEE Trans. on Computers*, Vol. C-19, No. 6, pp. 504-509, June 1970.

[15] S. Muroga, *Logic Design and Switching Theory*, Wiley-Interscience Publication, 1979.

[16] R. Rudell and A. Sangiovanni-Vincentelli, "Espresso-MV: Algorithm for Multiple-valued logic minimization," *Proc. IEEE Custom Integrated Circuit Conference (CICC)*, Portland, May 1985.

[17] T. Sasao, "Multiple-valued decomposition of generalized Boolean functions and the complexity of programmable logic arrays," *IEEE Trans. on Computers*, Vol. C-30, No. 9, pp. 635-643, Sept. 1981.

[18] T. Sasao, "On the optimal design of multiple-valued PLA's," *IEEE Trans. on Computers*, Vol. C-38, No. 4, pp. 582-592, April 1989.

[19] T. Sasao and M. Fujita (ed.), *Representation of Discrete Functions*, Kluwer Academic Publishers (1996-03).

[20] W. R. Smith III, *Minimization of multiple-valued functions*, in *Computer Science and Multiple-Valued Logic*, (e.d. D. C. Rine), North Holland, New York, 1984, pp. 227-267.

[21] A. Svoboda and D. E. White, *Advanced Logical Circuit Design Techniques*, Garland Press, New York, 1979.

[22] S. Yang, "Logic synthesis and optimization benchmark user guide, version 3.0," *MCNC*, Jan. 1991.